Лабораторная работа №2

Операторы ветвлений и логические условия в JavaScript

1. **ОПЕРАТОРЫ ВЕТВЛЕНИЙ. Оператор if.** Иногда, в зависимости от условия, нужно выполнить различные действия. Для этого используется оператор if.

Оператор if («если») получает условие. Он вычисляет его, и если результат — true, то выполняет команду.

Если нужно выполнить более одной команды при выполнении условия, то они оформляются блоком кода в фигурных скобках (создать приведенный пример документа. Может быть использован Visual Studio Code или любой другой редактор. Сохранить файл с названием Пример2_1.html и открыть (запустить) его в любом браузере):

```
<!DOCTYPE html>
<html>
<body>
<script>
let year = prompt('В каком году появилась спецификация HTML5', '');
/* функция prompt создает окно для ввода текста, который передается в
переменную
*/
if (year != 2014)
{
alert('А вот..');
alert('..и неправильно!');
}
{
document.write(cars[i] + "<br>");
}
</script>
</body>
</html>
```

Посмотрите работу скрипта - т.е. если вы вводите число не равное 2014 программа вам выдает сообщение, что вы не правы

Преобразование к логическому типу

Оператор if вычисляет и преобразует условие к логическому типу.

!!! В логическом контексте число 0, пустая строка "", null и undefined, а также NaN являются false, остальные значения — true.

Например, такое условие никогда не выполнится:

if (0) { // 0 преобразуется к false }

А такое — выполнится всегда:

if (1) { // 1 преобразуется к true}

Вычисление условия в проверке if (year != 2011) может быть вынесено в отдельную переменную:

let cond = (year != 2011); // вернет true или false в зависимости от year

if (cond) {

//какие-то операторы

}

2. **Условие else**

Необязательный блок else («иначе») выполняется, если условие неверно (Пример2_2.html)::

```
let year = prompt('Введите год выхода HTML5', '');
if (year == 2014) {
alert('Да вы знаток!');
} else {
alert('А вот и неправильно!'); // любое значение, кроме 2014
}
```

3. **Несколько условий, else if**

Бывает нужно проверить несколько вариантов условия. Для этого используется блок else if (Пример2_3.html)::

```
let year = prompt('В каком году появилась спецификация HTML5?', '');
if (year < 2014) {
alert('Это слишком рано..');
} else if (year > 2014) {
alert('Это поздновато..');
} else {
alert('Да, точно в этом году!');
}
```

4. Оператор вопросительный знак „?“

Иногда нужно в зависимости от условия присвоить переменную. Например (Пример2_4.html)::

```
let access;
let age = prompt('Сколько вам лет?', '');

if (age > 14) {
  access = true;
} else {
  access = false;
}
alert(access);
```

Оператор вопросительный знак '?' позволяет делать это короче и проще. Он состоит из трех частей:
условие ? значение1 : значение2
Проверяется условие, затем если оно верно – возвращается значение1, если неверно – значение2, например:

```
access = (age > 14) ? true : false;
```

Оператор '?' выполняется позже большинства других, в частности – позже сравнений, поэтому скобки можно не ставить:

```
access = age > 14 ? true : false;
```

…Но когда скобки есть – код лучше читается. Так что рекомендуется их писать.

5. Несколько операторов „?“

Последовательность операторов '?' позволяет вернуть значение в зависимости не от одного условия, а от нескольких.
Например:

```
let age = prompt('возраст?', 18);
let message = (age < 3) ? 'Здравствуй, малыш!' :
  (age < 18) ? 'Привет!' :
  (age < 100) ? 'Здравствуйте!' :
  'Какой необычный возраст!';
alert( message );
```

Поначалу может быть сложно понять, что происходит. Однако, внимательно приглядевшись, мы замечаем, что это обычная последовательная проверка!
Вопросительный знак проверяет сначала age < 3, если верно – возвращает 'Здравствуй, малыш!', если нет – идет за двоеточие и проверяет age < 18. Если это верно – возвращает 'Привет!', иначе проверка age < 100 и 'Здравствуйте!'… И наконец, если ничего из этого не верно, то 'Какой необычный возраст!'.
То же самое через if..else:

```
if (age < 3) {
  message = 'Здравствуй, малыш!';
} else if (age < 18) {
  message = 'Привет!';
} else if (age < 100) {
```

```
  message = 'Здравствуйте!';
} else {
  message = 'Какой необычный возраст!';
}
```

Нетрадиционное использование „?“

Иногда оператор вопросительный знак '?' используют как замену if:

```
let company = prompt('Какая компания создала JavaScript?', '');
(company == 'Netscape') ?
   alert('Да, верно') : alert('Неправильно');
```

Работает это так: в зависимости от условия, будет выполнена либо первая, либо вторая часть после'?'.

Результат выполнения не присваивается в переменную, так что пропадёт (впрочем, alert ничего не возвращает).

Рекомендуется не использовать вопросительный знак таким образом.

Несмотря на то, что с виду такая запись короче if, она является существенно менее читаемой.

Вот, для сравнения, то же самое с if:

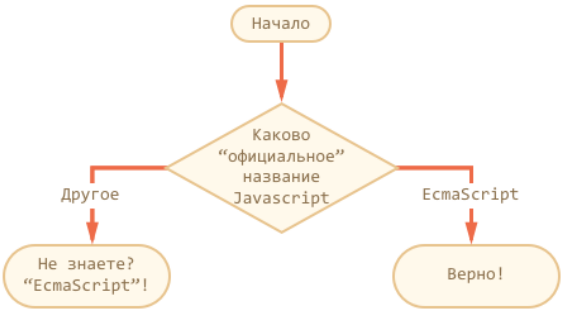
```
let company = prompt('Какая компания создала JavaScript?', '');

if (company == 'Netscape') {
  alert('Да, верно');
} else {
  alert('Неправильно');
}
```

При чтении кода глаз идёт вертикально и конструкции, занимающие несколько строк, с понятной вложенностью, воспринимаются гораздо легче. Возможно, вы и сами почувствуете, пробежавшись глазами, что синтаксис с if более прост и очевиден чем с оператором '?'.

Смысл оператора '?' – вернуть то или иное значение, в зависимости от условия. Пожалуйста, используйте его по назначению, а для выполнения разных веток кода есть if.

Самостоятельно:

Используя конструкцию if..else, напишите код, который будет спрашивать: «Каково «официальное» название JavaScript?». Если посетитель вводит «ECMAScript», то выводить «Верно!», если что-то другое – выводить «Не знаете? «ECMAScript»!».



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 1

Варианты распределяются по первой букве Вашей фамилии:

А…Е – 1 вариант;
Ж…М – 2 вариант;
Н…У – 3 вариант;
Ф…Я – 4 вариант

Вариант 1: Напишите код, который исправит предыдущий пример и будет отражать по введенной дате соответствующий номер декады месяца.

Вариант 2: Напишите код, который исправит предыдущий пример и будет отражать для введенной даты по условиям варианты веков в диапазоне (17 век, 18 век, 19 век, 20 век, 21 век), а для остальных случаев выводила бы сообщение «Век науке не известен».

Вариант 3: Напишите код, который получает значение prompt, а затем выводит alert:
«маловато», если сумма дня и номера месяца больше 1,

«все еще маловато», если сумма дня и номера месяца больше 10,
«терпимо», если сумма дня и номера месяца больше 20,
«сойдет», во всех остальных случаях.
Вариант 4: Напишите код, который исправит предыдущий пример и будет проверять високосный ли год. Если високосный, то выводить «ура, лишний день в году», иначе «все, как обычно».

Индивидуальное задание:
Вариант 1: Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?':

```
if (sqrt(a*b) < 4) {
result = 'Мало';
} else {
result = 'Много';
}
```

Вариант 2: Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?':

```
if (a==b) {
result = 'значения равны';
} else {
result = 'значения не равны';
}
```

Вариант 3: Напишите код, который получает значение трех переменных a, b и c через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?':

```
if ((a*a+b*b)=c*c) {
result = 'Ура: угадали длины сторон треугольника';
} else {
result = 'Надо учить теорему';
}
```

Вариант 4: Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?':

```
if ((a+a)==b) {
result = 'значения равны';
} else {
result = 'значения не равны';
}
```

Пример2_5.html:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Пример 2_3</title>
</head>
<body>
<script>
//объявляем массив как экземпляр класса Array
let month_names = new Array("January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December");
//создаем экземпляр класса Date
let d = new Date();
//создаем переменную и получаем день месяца из экземпляра d
let current_date = d.getDate();
```

```
//создаем переменную и получаем номер месяца из экземпляра d
//при чем счет месяцев идет с нуля
let current_month = d.getMonth();
//создаем переменную и получаем год в виде 4х чисел
let current_year = d.getFullYear();
alert("day - "+current_date+" ; month - " +current_month + " ; year - "+current_year);
if (current_month ==0) {
alert('На дворе январь'+ " ; "+ month_names[current_month]);
}
else if (current_month == 1) {
alert(' На дворе февраль'+ " ; "+ month_names[current_month]);
}
else {
alert('Месяц науке не известен!');
}
</script>
</body>
</html>
```

5. ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

В JavaScript поддерживаются операторы || (ИЛИ), && (И) и ! (НЕ).

Они называются «логическими», но в JavaScript могут применяться к значениям любого типа и возвращают также значения любого типа.

5.1. || (ИЛИ)

Оператор ИЛИ выглядит как двойной символ вертикальной черты:

```
result = a || b;
```

Логическое ИЛИ в классическом программировании работает следующим образом:

«если хотя бы один из аргументов true, то возвращает true, иначе — false».

Получается следующая таблица результатов:

```
alert( true || true ); // true
alert( false || true ); // true
alert( true || false); // true
alert( false || false); // false
```

При вычислении ИЛИ в JavaScript можно использовать любые значения. В этом случае они будут интерпретироваться как логические.

Например, число 1 будет воспринято как true, а 0 — как false:

```
if ( 1 || 0 ) { // сработает как if( true || false )
alert('верно');
}
```

Обычно оператор ИЛИ используется в if, чтобы проверить, выполняется ли хотя бы одно из условий, например:

```
let hour = 9;
if (hour < 10 || hour > 18) {
alert('Офис до 10 или после 18 закрыт');
}
```

Можно передать и больше условий:

```
let hour = 12, isWeekend = true;
if (hour < 10 || hour > 18 || isWeekend) {
alert('Офис до 10 или после 18 или в выходной закрыт');
}
```

Пример кода в составе страницы HTML. Пример2_6.html:

```
<!DOCTYPE HTML>
```

```
<html>
<head>
<title>Пример 2_5</title>
</head>
<body>
<script>
let hour = prompt('Введите интересующий час работы');
let isWeekend = prompt('Сегодня выходной? (да – true; нет - false)');
if (hour < 10 || hour > 18 || isWeekend) {
alert('Офис до 10 или после 18 или в выходной закрыт');
}
</script>
</body>
</html>
```

Короткий цикл вычислений

JavaScript вычисляет несколько ИЛИ слева направо. При этом, чтобы экономить ресурсы, используется так называемый «короткий цикл вычисления».

Допустим, вычисляются несколько ИЛИ подряд: a || b || c || .... Если первый аргумент — true, то результат заведомо будет true (хотя бы одно из значений — true), и остальные значения игнорируются.

Это особенно заметно, когда выражение, переданное в качестве второго аргумента, имеет сторонний эффект — например, присваивает переменную.

При запуске примера ниже присвоение x не произойдёт:

```
let x;
true || (x = 1); // просто вычислим ИЛИ, без if
alert(x); // undefined, x не присвоен
```

…А в примере ниже первый аргумент — false, так что ИЛИ попытается вычислить второй, запустив тем самым присваивание:

```
let x;
false || (x = 1);
alert(x); // 1
```

5.2. && (И)

Оператор И пишется как два амперсанда &&:

```
result = a && b;
```

В классическом программировании И возвращает true, если оба аргумента истинны, а иначе — false

```
alert( true && true ); // true
alert( false && true ); // false
alert( true && false); // false
alert( false && false); // false
```

Пример:

```
let hour = 12, minute = 30;
if (hour == 12 && minute == 30) {
alert('Время 12:30');
}
```

К И применим тот же принцип «короткого цикла вычислений», но немного по-другому, чем к ИЛИ. Если левый аргумент — false, оператор И возвращает его и заканчивает вычисления. Иначе — вычисляет и возвращает правый аргумент.

```
// Первый аргумент - true,
// Поэтому возвращается второй аргумент
alert(1 && 0); // 0
```

```
alert(1 && 5); // 5
// Первый аргумент - false,
// Он и возвращается, а второй аргумент игнорируется
alert(null && 5); // null
alert(0 && "не важно"); // 0
```

!!! Приоритет оператора И && больше, чем ИЛИ ||, т.е. он выполняется раньше.

Поэтому в следующем коде сначала будет вычислено правое И: 1 && 0 = 0, а уже потом — ИЛИ.

5.3. ! (НЕ)

Оператор НЕ — самый простой. Он получает один аргумент. Синтаксис:

```
let result = !value;
```

Действия !:

1. Сначала приводит аргумент к логическому типу true/false.
2. Затем возвращает противоположное значение.

Например:

```
alert( !true ) // false
alert( !0 ) // true
```

В частности, двойное НЕ используются для преобразования значений к логическому типу:

```
alert( !!"строка" ) // true
alert( !!null ) // false
```

Индивидуальное задание 2:

Вариант 1: Напишите код, который получает значение возраста и пола (с проверкой условия, что вводится только «м» или «ж»). Напишите условие if для проверки того факта, что переменная с возрастом находится между 14 и 25 включительно и если пол мужской, то выводить фраза «Добрый день, молодой человек», если тот же возраст, но пол женский, то выводить фразу «Приветствуем юную даму». Во всех остальных случаях выводить «Привет!».

Вариант 2: Напишите код, который получает должность (с проверкой, что должность введена «начальник», «зам.начальника» или «секретарь») и ФИО. Напишите условие if, если должность не «секретарь», то выводить «Добрый день, господин начальник», иначе если должность «секретарь», то выводить «Добрый день, ФИО».

Вариант 3: Напишите код, который получает значения для трех переменных: цвет, форма, размер. При этом предлагается пользователю в сообщении вводить для цвета - «синий» или «зеленый» или «красный», для формы - «круглый», «квадратный», для размера – «большой», «маленький». Напишите условие, если цвет не синий, а форма круглая и размер маленький, то выводить фразу «Наверное вы подумали про яблоко», если форма квадратная, а цвет любой, то выводить фразу «Наверное вы подумали про нечто квадратное», иначе «Понятия не имею, о чем вы думали».

Вариант 4: Напишите код, который получает значения для трех переменных: цвет, форма, размер. При этом предлагается пользователю в сообщении вводить для цвета - «синий» или «зеленый» или «красный», для формы - «круглый», «квадратный», для размера – «большой», «маленький». Напишите условие, если цвет не красный и не зеленый, а форма круглая и размер большой, то выводить фразу «Что-то синее и большое», если форма квадратная, а цвет не синий, то выводить фразу «Наверное вы подумали про яблоко, но почему квадратное?», иначе выводить значения переменных в одном сообщении.

Отчет по лабораторной работе

В соответствии со структурой заготовки отчета и примером оформления оформить в отчете все задания, выполняемые в ходе лабораторной работы, а также индивидуальные задания по вариантам. Файл с отчетом называть по шаблону: Фамилия_лаб_раб_номер.

Отчет предоставляется в электронном виде вместе с файлами задания и размещается на GitHabe либо присылается на электронную почту для проверки. Также по результатам лабораторной работы на следующем за ней занятии проводится выборочный опрос по командам языка.